

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A method of semaphoring between a system firmware and Advanced Configuration and Power Interface (ACPI) subsystem, the method comprising:

prior to entering a critical section of a shared memory by a first entity, checking, by the first entity, a turn flag to determine if the first entity or a second entity has a turn to access the critical section of the shared memory, wherein the turn flag ~~indicates~~ has a first value if the first entity has the turn to access the critical section ~~or~~ and wherein the turn flag has a second value if the second entity has the turn to access the critical section;

if the first entity will attempt to access the critical section and if the first entity has the turn to access the critical section of the shared memory, then changing, by the first entity, a value in the turn flag from the first value to the second value so that the second entity will have the turn to access the critical section of the shared memory;

if the first entity will attempt to access the critical section and if the second entity has the turn to access the critical section of the shared memory, then checking, by the first entity, an a second In flag of that is associated with the second entity to so that the first entity determines if the second entity is in the critical section, wherein the second In flag of the second entity has a the first value if the second entity is in the critical section and has the

second value if the second entity is not in the critical section;

if the first entity will attempt to access the critical section and if the second entity is in the critical section, then avoiding to enter the critical section by the first entity until after the second entity exits the critical section;

changing, by the second entity, the second In flag to the second value after the second entity exits the critical section;

changing, by the first entity, a first In flag that is associated with the first entity to the first value prior to the first entity entering the critical section, wherein the first In flag has the first value if the first entity is in the critical section and has the second value if the first entity is not in the critical section; and

entering the critical section by the first entity after the second entity exits the critical section and after the first entity has set the first In flag to the first value,
~~wherein the In flag has a second value if the second entity is not in the critical section.~~

Claim 2 (currently amended): The method of claim 1, further comprising:

setting ~~an~~ the first In flag ~~of the first entity~~ to the first value prior to the first entity entering the critical section and wherein the second entity does not enter the critical section when the first In flag has the first value.

Claim 3 (currently amended): The method of claim 1, further comprising:

setting ~~an~~ the first In flag ~~of the first entity~~ to the second value after ~~exiting~~ the first entity exits the critical section.

Claim 4 (Original): The method of claim 1, further comprising:

if the second entity is not in the critical section, then entering the critical section by the first entity.

Claim 5 (Original): The method of claim 1, wherein the first entity is the system firmware and the second entity is the ACPI subsystem.

Claim 6 (Original): The method of claim 1, wherein the first entity is the ACPI subsystem and the second entity is the system firmware.

Claim 7 (Original): The method of claim 1, wherein the first entity checks a shared memory to determine priority in the critical section.

Claim 8 (currently amended): An apparatus for semaphoring between a system firmware and Advanced Configuration and Power Interface (ACPI) subsystem, the apparatus comprising:

a first entity;
a second entity;
a shared memory;

a semaphoring system between the first entity and second entity, wherein prior to the first entity entering a critical section of the shared memory, wherein the first entity is configured to check a turn flag to determine if the first entity or the second entity has a turn to access a the critical section of the shared memory, wherein the turn flag indicates has a first value if the first entity has the turn to access the critical section or and wherein the turn flag has a second value if the second entity has the turn to access the critical section;

wherein if the first entity will attempt to access the critical section and if the first entity has the turn to access the critical section of the shared memory, then the first entity changes a value in the turn flag from the first value to the second value so that the second entity will have the turn to access the critical section of the shared memory;

wherein if the first entity will attempt to access the critical section, then the first entity is configured to check an a second In flag of that is associated with the second entity to so that the first entity determines if the second entity is in the critical section, if the second entity has the turn to access the critical section, 7 wherein the second In flag of the second entity has a the first value if the second entity is in the critical section and has the second value if the second entity is not in the critical section;

wherein the second entity changes the second In flag to the second value after the second entity exits the critical section;

wherein the first entity changes a first In flag that is associated with the first entity to the first value prior to the first entity entering the critical section, wherein the first In flag has the first value if the first entity is in the critical section and has the second value if the first entity is not in the critical section;

wherein the first entity is configured to avoid to enter the critical section until after the second entity exits the critical section, and to enter the critical section after exit of the critical section by the second entity and after the first entity has set the first In flag to the first value, ~~wherein the In flag has a second value if the second entity is not in the critical section.~~

Claim 9 (Original): The apparatus of claim 8, wherein the first entity is the system firmware and the second entity is the ACPI subsystem.

Claim 10 (Original): The apparatus of claim 8, wherein the first entity is the ACPI subsystem and the second entity is the system firmware.

Claim 11 (Original): The apparatus of claim 8, wherein the first entity checks a shared memory to determine priority in the critical section.

Claim 12 (currently amended): An apparatus for semaphoring between a system firmware and Advanced Configuration and Power Interface (ACPI) subsystem, the apparatus comprising:

means for checking a turn flag to determine if a first entity or a second entity has a turn to access a critical section of a shared memory, prior to entering the critical section by a first entity, wherein the turn flag ~~indicates~~ has a first value if the first entity has the turn to access the critical section or and wherein the turn flag has a second value if the second entity has the turn to access the critical section;

means for changing a value in the turn flag from the first value to the second value so that the second entity will have the turn to access the critical section of the shared memory if the first entity has the turn to access the critical section of the shared memory and if the first entity will attempt to access the critical section;

means for checking ~~an~~ a second In flag of that is associated with the second entity to so that the first entity determines if the second entity is in the critical section of the shared memory, if the second entity has the turn to access the critical section and if the first entity will attempt to access the critical section, wherein the second In flag of the second entity has a the first value if the second entity is in the critical section and has the second value if the second entity is not in the critical section;

means for avoiding to enter the critical section by the first entity until after the second entity ~~to~~ exits the critical section;

means for changing, by the second entity, the second In flag to the second value after the second entity exits the critical section;

means for changing, by the first entity, a first In flag that is associated with the first entity to the first value prior to the first entity entering the critical section, wherein the first In flag has the first value if the first entity is in the critical section and has the second value if the first entity is not in the critical section; and

means for entering the critical section by the first entity after the second entity exits the critical section and after the first entity has set the first In flag to the first value, ~~wherein the In flag has a second value if the second entity is not in the critical section.~~

Claim 13 (currently amended): An article of manufacture, comprising:

a machine-readable storage medium having stored thereon instructions to permit a computer to perform a method comprising:

prior to entering a critical section by a first entity, checking a turn flag to determine if the first entity or a second entity has a turn to access the critical section of the shared memory, wherein the turn flag has a first value if the first entity has the turn to access the critical section ~~or~~ and wherein the turn flag has a second value if the second entity has the turn to access the critical section;

if the first entity will attempt to access the critical section and if the first entity has the turn to access the critical section of the shared memory, then changing, by the first entity, a value in the turn flag from the first value to the second value so that the second entity will have the turn to access the critical section of the shared memory;

if the first entity will attempt to access the critical section and if the second entity has the turn to access the critical section of the shared memory, then checking ~~an~~ a second In flag ~~of~~ that is associated with the second entity ~~to~~ so that the first entity determines if the second entity is in the critical section, wherein the second In flag ~~of the second entity~~ has a the first value if the second entity is in the critical section and has the second value if the second entity is not in the critical section;

if the first entity will attempt to access the critical section and if the second entity is in the critical section, then avoiding to enter the critical section by the first entity until after the second entity exits the critical section;

changing, by the second entity, the second In flag to the second value after the second entity exits the critical section;

changing, by the first entity, a first In flag that is associated with the first entity to the first value prior to the first entity entering the critical section, wherein the first In flag has the first value if the first entity is in the critical section and has the second value if the first entity is not in the critical section; and

entering the critical section by the first entity after the second entity exits the critical section and after the first entity has set the first In flag to the first value,
~~wherein the In flag has a second value if the second entity is not in the critical section.~~

Claim 14 (currently amended): The apparatus of claim 8, wherein the first entity sets ~~an~~ the first In flag ~~of the first entity~~ to the first value prior to the first entity entering the critical section and wherein the second entity does not enter the critical section when the first In flag has the first value ~~by the first entity.~~

Claim 15 (currently amended): The apparatus of claim 8, wherein the first entity sets ~~an~~ the first In flag ~~of the first entity~~ to the second value after the first entity exits the critical section.

Claim 16 (currently amended): The apparatus of claim 12, wherein the first entity sets ~~an~~ the first In flag ~~of the first entity~~ to the first value prior to entering the critical section and wherein the second entity does not enter the critical section when the first In flag has the first value ~~by the first entity.~~

Claim 17 (currently amended): The apparatus of claim 12, wherein the first entity sets ~~an~~ the first In flag ~~of the first entity~~ to the second value after the first entity exits the critical section.

Claim 18 (currently amended): The article of manufacture of claim 13, wherein the first entity sets ~~an~~ the first In flag ~~of the first entity~~ to the first value prior to the first entity entering the critical section and wherein the second

entity does not enter the critical section when the first In flag has the first value ~~by the first entity.~~

Claim 19 (currently amended): The article of manufacture of claim 13, wherein the first entity sets ~~an~~ the In flag ~~of the first entity~~ to the second value after the first entity exits the critical section.